IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Tuan Huu Pham et al.	Art Unit	: 2194
Serial No.	: 09/855,683	Examiner	: Andy Ho
Filed	: May 16, 2001	Conf. No.	: 1532
Title	: COMPONENT INSTALLATION TOOL		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF (37 C.F.R 41.37)

In response to the notification of non-compliant Appeal Brief dated November 13, 2006, Applicant submits herewith a revised Appeal Brief in compliance with the requirements of 37 CFR § 41.37. The brief fee has previously been paid and, therefore, no fees are believed to be due at this time.

Please apply any charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: 11/28/06

Dmitry Brant
Reg. No. 59,133

Fish & Richardson P.C.
1425 K Street, N.W.
11th Floor
Washington, DC 20005-3500
Telephone: (202) 783-5070
Facsimile: (202) 783-2331

40380913.doc

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Tuan Huu Pham et al.	Art Unit	: 2194
Serial No.	: 09/855,683	Examiner	: Andy Ho
Filed	: May 16, 2001	Conf. No.	: 1532
Title	: COMPONENT INSTALLATION TOOL		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

TABLE OF CONTENTS

I. Real Party in Interest

America Online, Inc., the assignee of this application, is the real party is interest.

II. Related Appeals and Interferences

There are no related appeals or interferences.

III. Status of Claims

Claims 1, 8-14, 17, 18, 23-27, 30-32, 39-42, 45, 46, 51-55, 58-60, 67-70, 73, 74, 79-83, and 86-139 are pending and stand rejected, with claims 1, 32, 60, 88, 105, and 120 being independent.

Claims 2-7, 15, 16, 19-22, 28, 29, 33-38, 43, 44, 47-50, 56, 57, 61-66, 71, 72, 75-78, 84, and 85 are cancelled.

Appellants have appealed the rejection of claims 1, 8-14, 17, 18, 23-27, 30-32, 39-42, 45, 46, 51-55, 58-60, 67-70, 73, 74, 79-83, and 86-139.

IV. Status of Amendments

None.

V. Summary of Claimed Subject Matter

The following summarizes disclosure related to each independent claim with references to the application specification and drawings. The references to the specification and drawings are meant to be exemplary, and not limiting.

Independent claim 1

Independent claim 1 is directed to a system for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices. As shown in Figs. 6a and 6b, a component installation tool 622 includes a receiving module 6221, a detection module 6222, and an installation module 6223, to receive, detect and install connectivity components on a client device 620 to enable connectivity to a host system 610. See, e.g., specification, p. 17, line 12 to p. 18, line 13.

The receiving module 6221 is structured and arranged to receive a connectivity component that enables connectivity to a host system 610 by at least one of several different hardware devices. See, e.g., specification, p. 18, lines 15-18. The detection module 6222 is structured and arranged to detect whether installation of the connectivity component is needed to enable connectivity between the client device 620 and the host system 610 using a selected hardware device. See, e.g., specification, p. 18, lines 18-21. The installation module 6223 is structured and arranged to install the connectivity component when the connectivity component is needed to enable connectivity between the client device 620 and the host system 610 using the selected hardware device. See, e.g., specification, p. 18, lines 22-25.

More specifically, the detection module 6222 is structured and arranged to detect a new hardware device and, based on detecting the new hardware device, to determine whether a connectivity component is stored locally that is needed to enable connectivity between the client device 620 and the host system 610 using the new hardware device. See, e.g., specification, p. 20, lines 26-29. The receiving module 6221 is structured and arranged to receive an updated connectivity component from a remote server when the detection module 6222 determines that the connectivity component that is needed to enable connectivity between the client device 620 and the host system 610 using the new hardware device is not stored locally and the installation

module 6223 installs the updated connectivity component. See, e.g., specification, p. 20, line 31 to p. 21, line 2.

Independent claim 32

Independent claim 32 is directed to a method for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices. As shown in Fig. 7, the method includes receiving a connectivity component that enables connectivity to a host system by at least one of several different hardware devices (step 710), detecting whether installation of the connectivity component is needed to enable connectivity between the client device and the host system using a selected hardware device (step 720), and installing the connectivity component when connectivity component is needed to enable connectivity between the client device and the host system using the selected hardware device (step 730). See, e.g., specification, p. 22, lines 18-26.

The method also includes detecting a new hardware device and, based on detecting the new hardware device, determining whether a connectivity component is stored locally that is needed to enable connectivity between the client device and the host system using the new hardware device. When an updated connectivity component that is needed to enable connectivity is not stored locally, an updated connectivity component is received from a remote server and the updated connectivity component received from the remote server is installed. See, e.g., specification, p. 25, lines 5-13.

Independent claim 60

Independent claim 60 is directed to a computer program for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices, where the computer program is stored on a computer readable medium or is a propagated signal. A component installation tool 622 includes one or more software components capable of functioning in the manner described below. The specification and figures refer to modules of the component installation tool, which may be implemented as software components. See, e.g., specification, p. 4, lines 19-20, p. 17, lines 12-15, and p. 28, line 26 to p. 29, line17. As shown in Figs. 6a and 6b, a component installation tool

622 includes a receiving module 6221, a detection module 6222, and an installation module 6223, to receive, detect and install connectivity components on a client device 620 to enable connectivity to a host system 610. See, e.g., specification, p. 17, line 12 to p. 18, line 13.

A receiving code segment (receiving module 6221) causes the computer to receive a connectivity component that enables connectivity to a host system 610 by at least one of several different hardware devices. See, e.g., specification, p. 18, lines 15-18. A detection code segment (detection module 6222) causes the computer to detect whether installation of the connectivity component is needed to enable connectivity between the client device 620 and the host system 610 using a selected hardware device. See, e.g., specification, p. 18, lines 18-21. An installation code segment (installation module 6223) causes the computer to install the connectivity component when the connectivity component is needed to enable connectivity between the client device 620 and the host system 610 using the selected hardware device. See, e.g., specification, p. 18, lines 22-25.

The detection code segment (detection module 6222) causes the computer to detect a new hardware device and, based on detecting the new hardware device, to determine whether a connectivity component is stored locally that is needed to enable connectivity between the client device 620 and the host system 610 using the new hardware device. See, e.g., specification, p. 20, lines 26-29. The receiving code segment (receiving module 6221) causes the computer to receive an updated connectivity component from a remote server when the detection code segment (detection module 6222) determines that the connectivity component that is needed to enable connectivity between the client device 620 and the host system 610 using the new hardware device is not stored locally and the installation code segment (installation module 6223) installs the updated connectivity component. See, e.g., specification, p. 20, line 31 to p. 21, line 2.

Independent claim 88

Independent claim 88 is directed to a system for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices. As shown in Figs. 6a and 6b, a component installation tool 622 includes a receiving module 6221, a detection module 6222, and an installation module 6223, to receive,

detect and install connectivity components on a client device 620 to enable connectivity to a host system 610. See e.g., specification, p. 17, line 12 to p. 18, line 13.

The receiving module 6221 is structured and arranged to receive multiple connectivity components that enable connectivity to a host system 610 by at least one of several different hardware devices, where the receiving module 6221 is structured and arranged to copy the connectivity components to the client device 620 from a compact disk and store the connectivity components in a dormant state on the client device 620. See, e.g., specification, p. 18, lines 26-30 and p. 19, lines 4-11.

The detection module 6222 is structured and arranged to detect whether installation of at least one of the connectivity components is needed to enable connectivity between the client device and the host system using a selected hardware device. See, e.g., specification, p. 19, lines 14-25. The installation module 6223 is structured and arranged to install the connectivity component when the connectivity component is needed to enable connectivity between the client device and the host system using the selected hardware device. See, e.g., specification, p. 18, lines 21-25.

Independent claim 105

Independent claim 105 is directed to a method for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices. As shown in Fig. 7, the method includes receiving multiple connectivity components that enable connectivity to a host system by at least one of several different hardware devices, where receiving the connectivity components includes copying the connectivity components to the client device from a compact disk and storing the connectivity components in a dormant state on the client device (step 710). See, e.g., specification, p. 23, lines 15-25. The method includes detecting whether installation of at least one of the connectivity components is needed to enable connectivity between the client device and the host system using a selected hardware device (step 720) and installing the connectivity component when connectivity component is needed to enable connectivity between the client device and the host system using the selected hardware device (step 730). See, e.g., specification, p. 22, lines 18-26 and p. 23, line 15 to p. 24, line 6.

Independent claim 120

Independent claim 120 is directed to a computer program for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices, where the computer program is stored on a computer readable medium or is a propagated signal. A component installation tool 622 includes one or more software components capable of functioning in the manner described below. The specification and figures refer to modules of the component installation tool, which may be implemented as software components. See, e.g., specification, p. 4, lines 19-20, p. 17, lines 12-15, and p. 28, line 26 to p. 29, line17. As shown in Figs. 6a and 6b, a component installation tool 622 includes a receiving module 6221, a detection module 6222, and an installation module 6223, to receive, detect and install connectivity components on a client device 620 to enable connectivity to a host system 610. See, e.g., specification, p. 17, line 12 to p. 18, line 13.

A receiving code segment (receiving module 6221) causes the computer to receive multiple connectivity components that enable connectivity to a host system 610 by at least one of several different hardware devices, where the receiving code segment (receiving module 6221) causes the computer to copy the connectivity components to the client device 620 from a compact disk and store the connectivity components in a dormant state on the client device 620. See, e.g., specification, p. 18, lines 26-30 and p. 19, lines 4-11.

A detection code segment (detection module 6222) causes the computer to detect whether installation of at least one of the connectivity components is needed to enable connectivity between the client device and the host system using a selected hardware device. See, e.g., specification, p. 19, lines 14-25. An installation code segment (installation module 6223) causes the computer to install the connectivity component when the connectivity component is needed to enable connectivity between the client device and the host system using the selected hardware device. See, e.g., specification, p. 18, lines 21-25.

VI. Grounds of Rejection

A. Claims 1, 8-14, 23-27, 30-32, 39-42, 51-55, 58-60, 67-70, 79-83, 86-87, and 137-139 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Krishnan (U.S. Patent No. 6,075,863) in view of Onosaka (U.S. Patent No. 5,961,608) and further in view of Perlman (U.S. Patent No. 6,023,585).

B. Claims 88-91, 93-94, 97, 99-108, 110-111, 114, 116-123, 125-126, 129, and 131-136 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Krishnan in view of Onosaka and further in view of Elg (U.S. Patent No. 6,694,354).

C. Dependent claims 17, 18, 45, 46, 73, and 74 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Krishnan in view of Onosaka and Perlman and further in view of Coutts (U.S. Patent No. 6,311,165).

D. Dependent claims 95-96, 112-113, and 127-128 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Krishnan in view of Onosaka and Elg and further in view of Coutts.

VII. Argument

A. Claims 1, 8-14, 23-27, 30-32, 39-42, 51-55, 58-60, 67-70, 79-83, 86-87, and 137-139 would not have been obvious over Krishnan in view of Onosaka and further in view of Perlman.

Appellants request reversal of the rejection because Krishnan, Onosaka, and Perlman, either alone or combined as proposed, fail to describe or suggest these features. Specifically, neither Krishnan nor Onosaka nor Perlman describes or suggests detecting whether installation of the connectivity component is needed to enable connectivity between the client device and the host system using a selected hardware device, as recited in claims 1, 32, and 60.

Krishnan describes detecting whether devices ***communicating over an existing connection*** need an updated version of data encryption software, as opposed to detecting whether software is necessary ***to enable a connectivity*** between these devices:

> Loading an applet from a remote modem provides a facility for automatic data encryption services. For example, **any time a connection is established** between modem 10 and a similar remote modem, the use of a data encryption applet may be negotiated. If the modems already contain the same version of the encryption applet then data transfers may begin. If, however, one of the modems lacks the encryption applet, or has an older, out-of-date version, **the modems may negotiate to transfer the new version** (emphasis added). See Col. 4, line 62- Col. 5, line 3.

As evidenced in the paragraph above, Krishnan's modems may "negotiate to transfer the new version" of a data encryption applet when one of the modems lacks the correct version of the applet. However, before the above modems can even start **negotiating to transfer**, they must establish at least some rudimentary connectivity to each other. Without establishing initial connectivity, the modems simply cannot negotiate/transfer a new version of a data encryption applet.

Furthermore, the data encryption applet does not enable connectivity between modems, as required by claim 1. Instead of enabling connectivity, the data encryption applet allows modems to encrypt data traveling across the established connection. See Col. 3, lines 56-60. In

fact, Krishnan suggests that the negotiations for the new data encryption applet between the modems is completely independent of establishing a connection between modems:

> **The negotiations** may take place when a communications link is first established between modem 10 and the remote modem, **or may be delayed until secure communications are specifically requested** (emphasis added). See Col. 3, lines 56-63.

Thus, in Krishnan, the installation of the new data encryption applet does not actually enable connectivity between the client device and the host system using a selected hardware device. The new data encryption applet instead either enables security during communications or improves connection performance by providing the client with a newer/better version of the software. See Col. 5, line 33-41; Col. 5, lines 10-15.

In the last Final Office Action, the Examiner argued that the presence of the correct data encryption applet is necessary to enable connectivity, because if the encryption data applets "are not the same [on both modems], data transfer cannot occur." See Final Office Action mailed February 16, 2006, page 20. However, the Examiner appears to confuse the concept of "enabling connectivity" with the concept of enabling utilization of that which has been transferred over the enabled connection.

The data encryption applet enables security for data transferred over a pre-established connection, but does not enable connectivity between the modems themselves. In fact, the need for the data encryption applet presumes the existence of data to encrypt, and the existence of already established connection in Krishnan. Thus, in making this assertion, the Examiner seems to acknowledge precisely the point made above – that several prerequisites must exist to enable the data transfer described in Krishnan, among which is an existing connection over which an updated version could be received. Without debating whether those prerequisites include compatible encryption data applets as suggested by the Examiner, it is clear that the software transferred by Krishnan cannot be required to enable connectivity among the devices of Krishnan between which it is transferred. For, if the software were required to enable such connectivity, the connection required for its transfer between the devices of Krishnan would not exist.

This point is illustrated through reference to an example provided in Krishnan involving financial data transfer between a bank and a user's computer. See Krishnan, Col 3, lines 36-45.

In that example, the encryption applet provides application level encryption designed to secure data communications over an already-established or otherwise enabled connection between the bank's modem and the user's modem. So, while having the synchronized applets on both of the modems is important for communicating ***meaningful*** data across the connection, the connection between the modems must exist regardless of whether the encryption applet is available. Even if the correct applet were not available on either side of the connection, the data would still travel from the bank to the user's computer across a connection, although the user's computer would not know how to decrypt the incoming data.

The foregoing remarks focus on distinguishing Krishnan's system which is designed to upgrade software (e.g., data encryption applets) on a client device and which therefore requires an already established or otherwise enabled connection between that client device and the device from which needed software updates are received. To further distinguish the processes described by Krishnan and their focus on enabling a secure data transfer of application data over an existing or otherwise enabled connection, Applicant points out that claims 1, 32 and 50 recite enabled connectivity between the client device and the host system using a selected hardware device. No such teaching is provided by Krishnan.

To illustrate the potential significance of this distinction, an example is offered of a user who wishes to switch from a low-bandwidth connection type (e.g. an analog modem) to a broadband connection type (e.g. DSL/cable/satellite modem). See claims 12-14. In this example, the installation of a connectivity component (e.g. a device driver) for a selected hardware device, such as a cable modem, enables the user's computer to connect to the host system using the selected hardware device, thereby enabling faster connectivity. See e.g. Application, page 27, lines 4-14.

There also exists no motivation or suggestion to modify Krishnan with the teachings of Onosaka. The Final Office Action reasons that the convenience of using a suitable modem from multiple modems to connect to a remote computer, as taught by Onosaka, provides the motivation to modify Krishnan.

> "It would have been obvious to apply the teachings of Onosaka to the system of Krishnan because this gives the client computer the convenience of using a suitable modem from multiple modems to connect to a

remote computer as disclosed by Onosaka (lines 29-37 column 4)." See Final Office Action mailed February 16, 2006, pages 5.

However, this alleged motivation to modify Krishnan with Onosaka for reasons of convenience does not provide the required motivation or suggestion to modify Krishnan with the teachings of Onosaka. Since Krishnan is premised on the existence and prior installation of modems and their corresponding data encryption applets, the convenience of using a suitable modem from multiple modems is not relevant to Krishnan. Krishnan already has an existing modem to connect to a remote computer. Thus, there is no motivation or suggestion to modify Krishnan with Onosaka.

Additionally, even if a motivation existed to support a combination of Krishnan with Onosaka, the resulting combination would still not lead one of ordinary skill in the art to the invention of claim 1. Any possible combination of Krishnan with Onosaka must be premised on the respective teachings of those references taken as a whole without picking and choosing. As such, the Onosaka disclosure would allow a user to select a modem from a multitude of modems to update the encryption applet, such that integration of the Onasaka disclosure with the Krishnan disclosure would merely change the conduit through which the client device could update the encryption applet over an already-established or otherwise enabled connection. Moreover, the combination of Krishnan with Onosaka would fail to describe or suggest detecting whether installation of the connectivity component is needed to enable connectivity between the client device and the host system using a selected hardware device, as recited in claims 1, 32, and 60.

Similarly, there is no motivation or suggestion to modify Krishnan with the teachings of Perlman. The Final Office Action reasons that "it would have been obvious to apply the teachings of Perlman to the system of Krishnan because this allows the system to detect the new connected devices and downloading the appropriate software programs for these particular devices as disclosed by Perlman (lines 57-64, Col. 1)" See Final Office Action mailed February 16, 2006, pages 5-6.

However, this alleged motivation to modify Krishnan with Perlman for detection of newly connected devices does not provide the required motivation or suggestion to modify Krishnan with the teachings of Perlman. Since Krishnan is premised on the existence and prior

installation of modems on both sides of the connection, Krishnan does not envision addition of new modems and thus does not teach detecting new devices. Instead, Krishnan is concerned with maintenance of synchronized encryption applets on the host and client computers. In addition, the text cited by the Examiner to provide motivation for combining Krishnan and Perlman merely recites the "detecting a new hardware device" limitation, but does not provide any insight as to the motivation to combine Krishnan with Perlman. Thus, there is no motivation or suggestion to modify Krishnan with Perlman.

In addition, the teachings of Perlman, just like the teachings of Onosaka, fail to address the shortcomings of Krishnan. Specifically, Perlman focuses on automating the detection of new devices. Thus, even if a motivation existed to support a combination of Krishnan with Perlman, the resulting combination would still not lead one of ordinary skill in the art to the invention of claim 1. The combination of Krishnan and Perlman would, at most, suggest automatic detection of a device with which to update the data encryption applets, without regard for detecting whether installation of the connectivity component is needed to enable connectivity between the client device and the host system using a selected hardware device, as recited in claims 1, 32, and 60.

For at least these reasons, Appellants respectfully request reversal of the rejection of independent claims 1, 32, and 60, and their respective dependent claims.

B. Claims 88-91, 93-94, 97, 99-108, 110-111, 114, 116-123, 125-126, 129, and 131-136 would not have been obvious over Krishnan in view of Onosaka and further in view of Elg.

Appellants request reversal of these claims because Krishnan, Onosaka, or Elg either alone or in combination, fail to describe or suggest detecting whether installation of the connectivity component is needed to enable connectivity between the client device and the host system using a selected hardware device.

Neither Onosaka nor Elg remedy this failure of Krishnan and, notably, are not relied upon by the Final Office Action to describe or suggest these features.

Furthermore, there is no motivation or suggestion to modify Krishnan with the teachings of Onosaka, as discussed above.

Finally, there is no motivation or suggestion to modify Krishnan with the teachings of Elg. The Final Office Action reasons that "it would have been obvious to apply the teachings of Elg to the system of Krishnan because this allows the system to detect the new connected devices and downloading the appropriate software programs for these particular devices as disclosed by Elg (lines 15-35, Col. 1)" See Final Office Action mailed February 16, 2006, page 12.

However, this alleged motivation to modify Krishnan with Elg for detection of newly connected devices does not provide the required motivation or suggestion to modify Krishnan with the teachings of Elg. Since Krishnan is premised on the existence and prior installation of modems on both sides of the connection, Krishnan does not envision addition of new modems and thus does not teach detecting new devices. Instead, Krishnan is concerned with maintenance of synchronized encryption applets on the host and client computers. In addition, the text cited by the Examiner to provide motivation for combining Krishnan and Egl merely recites the "detecting a new hardware device" limitation, but does not provide any insight as to the motivation to combine Krishman with Elg. Thus, there is no motivation or suggestion to modify Krishnan with Elg.

In addition, the teachings of Elg, just like the teachings of Onosaka, fail to address the shortcomings of Krishnan because they focus on detection of new devices. Thus, even if a motivation existed to support a combination of Krishnan with Onosaka and Elg, the resulting combination would still not lead one of ordinary skill in the art to the invention of claim 1. The combination of Krishnan, Onasaka, and Elg would suggest detection of a new device with which to update the data encryption applets, without regard for detecting whether installation of the connectivity component is needed to enable connectivity between the client device and the host system using a selected hardware device, as recited in claims 1, 32, and 60.

For at least these reasons, Appellants respectfully request reversal of the rejection of independent claims 88, 105, and 120, and their respective dependent claims.

C. Dependent claims 17, 18, 45, 46, 73, and 74 would not have been obvious over Krishnan in view of Onosaka and Perlman and further in view of Coutts.

With respect to dependent claims 17, 18, 45, 46, 73, and 74, Appellants respectfully request reversal of the rejection because Krishnan, Onosaka, Perlman and Coutts, either alone or in combination, fail to describe or suggest the features discussed above with respect to the independent claims upon which these claims depend. Coutts does not remedy the failure of Krishnan, Onosaka, and Perlman to describe the features discussed above with respect to the independent claims and, notably, is not relied upon in the Final Office Action to support these features.

D. Dependent claims 95-96, 112-113, and 127-128 would not have been obvious over Krishnan in view of Onosaka and Elg and further in view of Coutts.

With respect to dependent claims 95-96, 112-113, and 127-128, Appellants respectfully request reversal of the rejection because Krishnan, Onosaka, Elg, and Coutts, either alone or in combination, fail to describe or suggest the features discussed above with respect to the independent claims upon which these claims depend. Coutts does not remedy the failure of Krishnan, Onosaka, and Elg to describe the features discussed above with respect to the independent claims and, notably, is not relied upon in the Final Office Action to support these features.

The brief fee has previously been paid and, therefore, no fees are believed to be due at this time. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: 10/22/06

Dmitry Brant
Reg. No. 59,133

Fish & Richardson P.C.
1425 K Street, N.W.
11th Floor
Washington, DC 20005-3500
Telephone: (202) 783-5070
Facsimile: (202) 783-2331

VIII. Claims Appendix

1. (previously presented) A system for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices, comprising:

a receiving module that is structured and arranged to receive a connectivity component that enables connectivity to a host system by at least one of several different hardware devices;

a detection module that is structured and arranged to detect whether installation of the connectivity component is needed to enable connectivity between the client device and the host system using a selected hardware device; and

an installation module that is structured and arranged to install the connectivity component when the connectivity component is needed to enable connectivity between the client device and the host system using the selected hardware device, wherein:

the detection module is structured and arranged to detect a new hardware device and, based on detecting the new hardware device, to determine whether a connectivity component is stored locally that is needed to enable connectivity between the client device and the host system using the new hardware device;

the receiving module is structured and arranged to receive an updated connectivity component from a remote server when the detection module determines that the connectivity component that is needed to enable connectivity between the client device and the host system using the new hardware device is not stored locally; and

the installation module is structured and arranged to install the updated connectivity component received from the remote server.

2-7. (canceled).

8. (previously presented) The system of claim 1 wherein the connectivity component is capable of interfacing a device driver to enable communications between computer software at

the client device and the host system using the new hardware device associated with the device driver.

9. (previously presented) The system of claim 1 wherein the connectivity component is capable of interfacing directly with the new hardware device to enable connectivity between the client device and the host system using the new hardware device.

10. (previously presented) The system of claim 1 wherein the connectivity component includes computer software that interfaces with a driver for the new hardware device that is used to connect to the host system.

11. (original) The system of claim 1 wherein the connectivity component includes a broadband connectivity component to enable connectivity to the host system using a broadband communication device.

12. (original) The system of claim 11 wherein the connectivity component includes a DSL connectivity component to enable connectivity to the host system using a DSL modem.

13. (original) The system of claim 11 wherein the connectivity component includes a cable connectivity component to enable connectivity to the host system using a cable modem.

14. (original) The system of claim 11 wherein the connectivity component includes a satellite connectivity component to enable connectivity to the host system using a satellite modem.

15. (canceled).

16. (canceled).

17. (original) The system of claim 1 wherein the installation module installs a list of programs needed to install the connectivity component.

18. (original) The system of claim 17 wherein the list of programs includes a sequential list of programs needed to install the connectivity component such that only one reboot of the client device is necessary to accomplish installation using the sequential list of programs.

19-22. (canceled).

23. (previously presented) The system of claim 1 wherein the receiving module, the detection module, and the installation module are structured and arranged to perform automatically without user intervention to receive, detect, and install the connectivity component to enable connectivity to the host system using the new hardware device.

24. (original) The system of claim 1 wherein the connectivity component received includes an updated version of a connectivity component stored on the client device before the connectivity component is installed by the installation module.

25. (original) The system of claim 24 wherein the detection module is structured and arranged to detect whether installation of the connectivity component is needed on the client device by comparing a version of the updated connectivity component received with a version of the connectivity component stored on the client device.

26. (original) The system of claim 1 wherein:

the receiving module is structured and arranged to receive version information from a remote server associated with a connectivity component;

the detection module is structured and arranged to detect whether installation of the connectivity component is needed on the client device, and to determine a version of the connectivity component to install by comparing the version information received from the

remote server with version information associated with the connectivity component already received by the receiving module when installation of the connectivity component is needed; and

the installation module is structured and arranged to install the connectivity component stored on the client device when the detection module determines the version information associated with the stored connectivity component is correct when compared against the version information received from the remote server.

27. (original) The system of claim 26 wherein:

the receiving module is structured and arranged to receive an updated connectivity component from the remote server when the detection module determines that the version information associated with the connectivity component stored on the client device is not correct when compared against the version information received from the remote server; and

the installation module is structured and arranged to install the updated connectivity component received from the remote server.

28. (canceled).

29. (canceled).

30. (original) The system of claim 1 wherein:

the receiving module is structured and arranged to include a host system receiving module that is structured and arranged to receive a request to send a connectivity component to a local client device; and

the installation module is structured and arranged to include a host system installation module that is structured and arranged to send the connectivity component to the local client device for installation on the local client device in response to the request.

31. (original) The system of claim 30 wherein the detection module is structured and arranged to include a host system detection module that is structured and arranged to determine a version of the connectivity component needed for installation on the local client device.

32. (previously presented) A method for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices, the method comprising:

receiving a connectivity component that enables connectivity to a host system by at least one of several different hardware devices;

detecting whether installation of the connectivity component is needed to enable connectivity between the client device and the host system using a selected hardware device; and

installing the connectivity component when connectivity component is needed to enable connectivity between the client device and the host system using the selected hardware device, wherein:

detecting whether installation of the connectivity component is necessary includes detecting a new hardware device and, based on detecting the new hardware device, determining whether a connectivity component is stored locally that is needed to enable connectivity between the client device and the host system using the new hardware device;

receiving the connectivity component includes receiving an updated connectivity component from a remote server when the connectivity component that is needed to enable connectivity is not stored locally; and

installing the connectivity component includes installing the updated connectivity component received from the remote server.

33-38. (canceled).

39. (previously presented) The method as in claim 32 wherein the connectivity component is capable of interfacing a device driver to enable communications between computer software at the client device and the host system using the new hardware device associated with the device driver.

40. (previously presented) The method as in claim 32 wherein the connectivity component is capable of interfacing directly with the new hardware device to enable connectivity between the client device and the host system using the new hardware device.

41. (previously presented) The method as in claim 32 wherein the connectivity component includes computer software that interfaces with a driver for the new hardware device that is used to connect to the host system.

42. (original) The method as in claim 32 wherein the connectivity component includes a broadband connectivity component to enable connectivity to the host system using a broadband communication device.

43. (canceled).

44. (canceled).

45. (original) The method as in claim 32 wherein installing the connectivity component includes installing a list of programs needed to install the connectivity component.

46. (original) The method as in claim 45 wherein the list of programs includes a sequential list of programs needed to install the connectivity component such that only one reboot of the client device is necessary to accomplish installation using the sequential list of programs.

47-50. (canceled).

51. (previously presented) The method as in claim 32 wherein receiving the connectivity component, detecting whether installation of the connectivity component is necessary, and installing the connectivity component includes automatically without user intervention receiving,

detecting, and installing the connectivity component to enable connectivity to the host system using the new hardware device.

52. (original) The method as in claim 32 wherein the connectivity component received includes an updated version of a connectivity component stored on the client device before the connectivity component is installed by the installation module.

53. (original) The method as in claim 52 wherein detecting whether installation of the connectivity component is needed includes comparing a version of the updated connectivity component received with a version of the connectivity component stored on the client device.

54. (original) The method as in claim 32 wherein:

receiving the connectivity component includes receiving version information from a remote server associated with a connectivity component;

detecting whether installation of the connectivity component is needed includes detecting whether installation of the connectivity component is needed on the client device, and determining a version of the connectivity component to install by comparing the version information received from the remote server with version information associated with the connectivity component already received when installation of the connectivity component is needed; and

installing the connectivity component includes installing the connectivity component stored on the client device when the version information associated with the stored connectivity component is determined to be correct when compared against the version information received from the remote server.

55. (original) The method as in claim 54 wherein:

receiving the connectivity component includes receiving an updated connectivity component from the remote server when the version information associated with the connectivity component stored on the client device is determined not to be correct when compared against the version information received from the remote server; and

installing the connectivity component includes installing the updated connectivity component received from the remote server.

56. (canceled).

57. (canceled).

58. (original) The method as in claim 32 wherein:

receiving the connectivity component includes using a host system to receive a request to send a connectivity component to a local client device; and

installing the connectivity component includes using the host system to send the connectivity component to the local client device for installation on the local client device in response to the request.

59. (original) The method as in claim 58 wherein detecting whether installation of the connectivity component is necessary includes using the host system to determine a version of the connectivity component to install on the local client device.

60. (previously presented) A computer program for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices, the computer program being stored on a computer readable medium or a propagated signal and comprising:

a receiving code segment that causes the computer to receive a connectivity component that enables connectivity to a host system by at least one of several different hardware devices;

a detection code segment that causes the computer to detect whether installation of the connectivity component is needed to enable connectivity between the client device and the host system using a selected hardware device; and

an installation code segment that causes the computer to install the connectivity component when the connectivity component is needed to enable connectivity between the client device and the host system using the selected hardware device, wherein:

the detection code segment causes the computer to detect a new hardware device and, based on detecting the new hardware device, to determine whether a connectivity component is stored locally that is needed to enable connectivity between the client device and the host system using the new hardware device;

the receiving code segment causes the computer to receive an updated connectivity component from a remote server when the detection code segment determines that the connectivity component that is needed to enable connectivity between the client device and the host system using the new hardware device is not stored locally; and

the installation code segment causes the computer to install the updated connectivity component received from the remote server.

61-66. (canceled).

67. (previously presented) The computer program of claim 60 wherein the connectivity component is capable of interfacing a device driver to enable communications between computer software at the client device and the host system using the new hardware device associated with the device driver.

68. (previously presented) The computer program of claim 60 wherein the connectivity component is capable of interfacing directly with the new hardware device to enable connectivity between the client device and the host system using the new hardware device.

69. (previously presented) The computer program of claim 60 wherein the connectivity component includes computer software that interfaces with a driver for the new hardware device that is used to connect to the host system.

70. (original) The computer program of claim 60 wherein the connectivity component includes a broadband connectivity component to enable connectivity to the host system using a broadband communication device.

71. (canceled).

72. (canceled).

73. (original) The computer program of claim 60 wherein the installation code segment causes the computer to install a list of programs needed to install the connectivity component.

74. (original) The computer program of claim 73 wherein the list of programs includes a sequential list of programs needed to install the connectivity component such that only one reboot of the client device is necessary to accomplish installation using the sequential list of programs.

75-78. (canceled).

79. (previously presented) The computer program of claim 60 wherein the receiving code segment, the detection code segment, and the installation code segment cause the computer to perform automatically without user intervention to receive, detect, and install the connectivity component to enable connectivity to the host system using the new hardware device.

80. (original) The computer program of claim 60 wherein the connectivity component received includes an updated version of a connectivity component stored on the client device before the connectivity component is installed by the installation code segment.

81. (original) The computer program of claim 80 wherein the detection code segment causes the computer to detect whether installation of the connectivity component is needed on the client device by comparing a version of the updated connectivity component received with a version of the connectivity component stored on the client device.

82. (original) The computer program of claim 60 wherein:

the receiving code segment causes the computer to receive version information from a remote server associated with a connectivity component;

the detection code segment causes the computer to detect whether installation of the connectivity component is needed on the client device, and to determine a version of the connectivity component to install by comparing the version information received from the remote server with version information associated with the connectivity component already received by the receiving code segment when installation of the connectivity component is needed; and

the installation code segment causes the computer to install the connectivity component stored on the client device when the detection code segment determines the version information associated with the stored connectivity component is correct when compared against the version information received from the remote server.

83. (original) The computer program of claim 82 wherein:

the receiving code segment causes the computer to receive an updated connectivity component from the remote server when the detection code segment determines that the version information associated with the connectivity component stored on the client device is not correct when compared against the version information received from the remote server; and

the installation code segment causes the computer to install the updated connectivity component received from the remote server.

84. (canceled).

85. (canceled).

86. (original) The computer program of claim 60 wherein:

the receiving code segment includes a host system receiving code segment that causes the computer to receive a request to send a connectivity component to a local client device; and

the installation code segment includes a host system installation code segment that causes the computer to send the connectivity component to the local client device for installation on the local client device in response to the request.

87. (original) The computer program of claim 86 wherein the host system detection code segment causes the computer to determine a version of the connectivity component to install on the local client device.

88. (previously presented) A system for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices, comprising:

a receiving module that is structured and arranged to receive multiple connectivity components that enable connectivity to a host system by at least one of several different hardware devices, wherein the receiving module is structured and arranged to copy the connectivity components to the client device from a compact disk and store the connectivity components in a dormant state on the client device;

a detection module that is structured and arranged to detect whether installation of at least one of the connectivity components is needed to enable connectivity between the client device and the host system using a selected hardware device; and

an installation module that is structured and arranged to install the connectivity component when the connectivity component is needed to enable connectivity between the client device and the host system using the selected hardware device.

89. (previously presented) The system of claim 88 wherein the connectivity component is capable of interfacing a device driver to enable communications between computer software at the client device and the host system using the selected hardware device associated with the device driver.

90. (previously presented) The system of claim 88 wherein the connectivity component is capable of interfacing directly with the selected hardware device to enable connectivity between the client device and the host system using the selected hardware device.

91. (previously presented) The system of claim 88 wherein the connectivity component includes computer software that interfaces with a driver for the selected hardware device that is used to connect to the host system.

92. (previously presented) The system of claim 88 wherein the connectivity component includes a broadband connectivity component to enable connectivity to the host system using a broadband communication device.

93. (previously presented) The system of claim 88 wherein:

the detection module is structured and arranged to detect whether the installation of the connectivity component is needed to enable connectivity between the client device and the host system in response to an input received from a user of the client device requesting communications using the selected hardware device; and

the installation module is structured and arranged to install the connectivity component when the connectivity component is needed based on the input from the user of the client device.

94. (previously presented) The system of claim 93 wherein the input from the user of the client device includes a request to change connectivity to the host system from a low-bandwidth connection type to a broadband connection type.

95. (previously presented) The system of claim 88 wherein the installation module installs a list of programs needed to install the connectivity component.

96. (previously presented) The system of claim 95 wherein the list of programs includes a sequential list of programs needed to install the connectivity component such that only one

reboot of the client device is necessary to accomplish installation using the sequential list of programs.

97. (previously presented) The system of claim 88 wherein the detection module further comprises an automatic hardware device detector that is structured and arranged to automatically detect a hardware device and determine the connectivity component needed to enable connectivity to the host system associated with the hardware device detected.

98. (previously presented) The system of claim 88 wherein the receiving module, the detection module, and the installation module are structured and arranged to perform automatically without user intervention to receive, detect, and install the connectivity component to enable connectivity to the host system using the selected hardware device.

99. (previously presented) The system of claim 88 wherein the connectivity component received includes an updated version of a connectivity component stored on the client device before the connectivity component is installed by the installation module.

100. (previously presented) The system of claim 99 wherein the detection module is structured and arranged to detect whether installation of the connectivity component is needed on the client device by comparing a version of the updated connectivity component received with a version of the connectivity component stored on the client device.

101. (previously presented) The system of claim 88 wherein:

the receiving module is structured and arranged to receive version information from a remote server associated with a connectivity component;

the detection module is structured and arranged to detect whether installation of the connectivity component is needed on the client device, and to determine a version of the connectivity component to install by comparing the version information received from the remote server with version information associated with the connectivity component already received by the receiving module when installation of the connectivity component is needed; and

the installation module is structured and arranged to install the connectivity component stored on the client device when the detection module determines the version information associated with the stored connectivity component is correct when compared against the version information received from the remote server.

102. (previously presented) The system of claim 101 wherein:

the receiving module is structured and arranged to receive an updated connectivity component from the remote server when the detection module determines that the version information associated with the connectivity component stored on the client device is not correct when compared against the version information received from the remote server; and

the installation module is structured and arranged to install the updated connectivity component received from the remote server.

103. (previously presented) The system of claim 88 wherein:

the receiving module is structured and arranged to include a host system receiving module that is structured and arranged to receive a request to send a connectivity component to a local client device; and

the installation module is structured and arranged to include a host system installation module that is structured and arranged to send the connectivity component to the local client device for installation on the local client device in response to the request.

104. (previously presented) The system of claim 103 wherein the detection module is structured and arranged to include a host system detection module that is structured and arranged to determine a version of the connectivity component needed for installation on the local client device.

105. (previously presented) A method for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices, the method comprising:

receiving multiple connectivity components that enable connectivity to a host system by at least one of several different hardware devices, wherein receiving the connectivity components includes copying the connectivity components to the client device from a compact disk and storing the connectivity components in a dormant state on the client device;

detecting whether installation of at least one of the connectivity components is needed to enable connectivity between the client device and the host system using a selected hardware device; and

installing the connectivity component when connectivity component is needed to enable connectivity between the client device and the host system using the selected hardware device.

106. (previously presented) The method as in claim 105 wherein the connectivity component is capable of interfacing a device driver to enable communications between computer software at the client device and the host system using the selected hardware device associated with the device driver.

107. (previously presented) The method as in claim 105 wherein the connectivity component is capable of interfacing directly with the selected hardware device to enable connectivity between the client device and the host system using the selected hardware device.

108. (previously presented) The method as in claim 105 wherein the connectivity component includes computer software that interfaces with a driver for the selected hardware device that is used to connect to the host system.

109. (previously presented) The method as in claim 105 wherein the connectivity component includes a broadband connectivity component to enable connectivity to the host system using a broadband communication device.

110. (previously presented) The method as in claim 105 wherein:

detecting whether installation of the connectivity component is necessary includes detecting whether the installation of the connectivity component is needed to enable connectivity

between the client device and the host system in response to an input received from a user of the client device requesting communications using the selected hardware device; and

installing the connectivity component includes installing the connectivity component when the connectivity component is needed based on the input from the user of the client device.

111. (previously presented) The method as in claim 110 wherein the input from the user of the client device includes a request to change connectivity to the host system from a low-bandwidth connection type to a broadband connection type.

112. (previously presented) The method as in claim 105 wherein installing the connectivity component includes installing a list of programs needed to install the connectivity component.

113. (previously presented) The method as in claim 112 wherein the list of programs includes a sequential list of programs needed to install the connectivity component such that only one reboot of the client device is necessary to accomplish installation using the sequential list of programs.

114. (previously presented) The method as in claim 105 wherein detecting whether installation of the connectivity component is necessary further comprises automatically detecting a hardware device and determining the connectivity component needed to enable connectivity to the host system associated with the hardware device detected.

115. (previously presented) The method as in claim 105 wherein receiving the connectivity component, detecting whether installation of the connectivity component is necessary, and installing the connectivity component includes automatically without user intervention receiving, detecting, and installing the connectivity component to enable connectivity to the host system using the selected hardware device.

116. (previously presented) The method as in claim 105 wherein:

receiving the connectivity component includes receiving version information from a remote server associated with a connectivity component;

detecting whether installation of the connectivity component is needed includes detecting whether installation of the connectivity component is needed on the client device, and determining a version of the connectivity component to install by comparing the version information received from the remote server with version information associated with the connectivity component already received when installation of the connectivity component is needed; and

installing the connectivity component includes installing the connectivity component stored on the client device when the version information associated with the stored connectivity component is determined to be correct when compared against the version information received from the remote server.

117. (previously presented) The method as in claim 116 wherein:

receiving the connectivity component includes receiving an updated connectivity component from the remote server when the version information associated with the connectivity component stored on the client device is determined not to be correct when compared against the version information received from the remote server; and

installing the connectivity component includes installing the updated connectivity component received from the remote server.

118. (previously presented) The method as in claim 105 wherein:

receiving the connectivity component includes using a host system to receive a request to send a connectivity component to a local client device; and

installing the connectivity component includes using the host system to send the connectivity component to the local client device for installation on the local client device in response to the request.

119. (previously presented) The method as in claim 118 wherein detecting whether installation of the connectivity component is necessary includes using the host system to determine a version of the connectivity component to install on the local client device.

120. (previously presented) A computer program for installing computer software components on a client device for enabling connectivity to a host system by at least one of several different hardware devices, the computer program being stored on a computer readable medium or a propagated signal and comprising:

a receiving code segment that causes the computer to receive multiple connectivity components that enable connectivity to a host system by at least one of several different hardware devices, wherein the receiving code segment causes the computer to copy at least one of the connectivity components to the client device from a compact disk and store the connectivity component in a dormant state on the client device;

a detection code segment that causes the computer to detect whether installation of the connectivity component is needed to enable connectivity between the client device and the host system using a selected hardware device; and

an installation code segment that causes the computer to install the connectivity component when the connectivity component is needed to enable connectivity between the client device and the host system using the selected hardware device.

121. (previously presented) The computer program of claim 120 wherein the connectivity component is capable of interfacing a device driver to enable communications between computer software at the client device and the host system using the selected hardware device associated with the device driver.

122. (previously presented) The computer program of claim 120 wherein the connectivity component is capable of interfacing directly with the selected hardware device to enable connectivity between the client device and the host system using the selected hardware device.

123. (previously presented) The computer program of claim 120 wherein the connectivity component includes computer software that interfaces with a driver for the selected hardware device that is used to connect to the host system.

124. (previously presented) The computer program of claim 120 wherein the connectivity component includes a broadband connectivity component to enable connectivity to the host system using a broadband communication device.

125. (previously presented) The computer program of claim 120 wherein:

the detection code segment causes the computer to detect whether the installation of the connectivity component is needed to enable connectivity between the client device and the host system in response to an input received from a user of the client device requesting communications using the selected hardware device; and

the installation code segment causes the computer to install the connectivity component when the connectivity component is needed based on the input from the user of the client device.

126. (previously presented) The computer program of claim 125 wherein the input from the user of the client device includes a request to change connectivity to the host system from a low-bandwidth connection type to a broadband connection type.

127. (previously presented) The computer program of claim 120 wherein the installation code segment causes the computer to install a list of programs needed to install the connectivity component.

128. (previously presented) The computer program of claim 127 wherein the list of programs includes a sequential list of programs needed to install the connectivity component such that only one reboot of the client device is necessary to accomplish installation using the sequential list of programs.

129. (previously presented) The computer program of claim 120 wherein the detection code segment further comprises an automatic hardware device detector code segment that causes the computer to automatically detect a hardware device and determine the connectivity component needed to enable connectivity to the host system associated with the hardware device detected.

130. (previously presented) The computer program of claim 120 wherein the receiving code segment, the detection code segment, and the installation code segment cause the computer to perform automatically without user intervention to receive, detect, and install the connectivity component to enable connectivity to the host system using the selected hardware device.

131. (previously presented) The computer program of claim 120 wherein the connectivity component received includes an updated version of a connectivity component stored on the client device before the connectivity component is installed by the installation code segment.

132. (previously presented) The computer program of claim 131 wherein the detection code segment causes the computer to detect whether installation of the connectivity component is needed on the client device by comparing a version of the updated connectivity component received with a version of the connectivity component stored on the client device.

133. (previously presented) The computer program of claim 120 wherein:

the receiving code segment causes the computer to receive version information from a remote server associated with a connectivity component;

the detection code segment causes the computer to detect whether installation of the connectivity component is needed on the client device, and to determine a version of the connectivity component to install by comparing the version information received from the remote server with version information associated with the connectivity component already received by the receiving code segment when installation of the connectivity component is needed; and

the installation code segment causes the computer to install the connectivity component stored on the client device when the detection code segment determines the version information associated with the stored connectivity component is correct when compared against the version information received from the remote server.

134. (previously presented) The computer program of claim 133 wherein:

the receiving code segment causes the computer to receive an updated connectivity component from the remote server when the detection code segment determines that the version information associated with the connectivity component stored on the client device is not correct when compared against the version information received from the remote server; and

the installation code segment causes the computer to install the updated connectivity component received from the remote server.

135. (previously presented) The computer program of claim 120 wherein:

the receiving code segment includes a host system receiving code segment that causes the computer to receive a request to send a connectivity component to a local client device; and

the installation code segment includes a host system installation code segment that causes the computer to send the connectivity component to the local client device for installation on the local client device in response to the request.

136. (previously presented) The computer program of claim 135 wherein the host system detection code segment causes the computer to determine a version of the connectivity component to install on the local client device.

137. (previously presented) The system of claim 1 wherein the detection module is structured and arranged to detect a prior receipt of the connectivity component that is needed to enable connectivity between the client device and the host system using the new hardware device.

138. (previously presented) The method as in claim 32 wherein detecting whether installation of the connectivity component is necessary includes detecting a prior receipt of the connectivity component that is needed to enable connectivity between the client device and the host system using the new hardware device.

139. (previously presented) The computer program of claim 60 wherein the detection code segment causes the computer to detect a prior receipt of the connectivity component that is needed to enable connectivity between the client device and the host system using the new hardware device

IX. Evidence Appendix

None.

X. Related Proceedings Appendix

None.